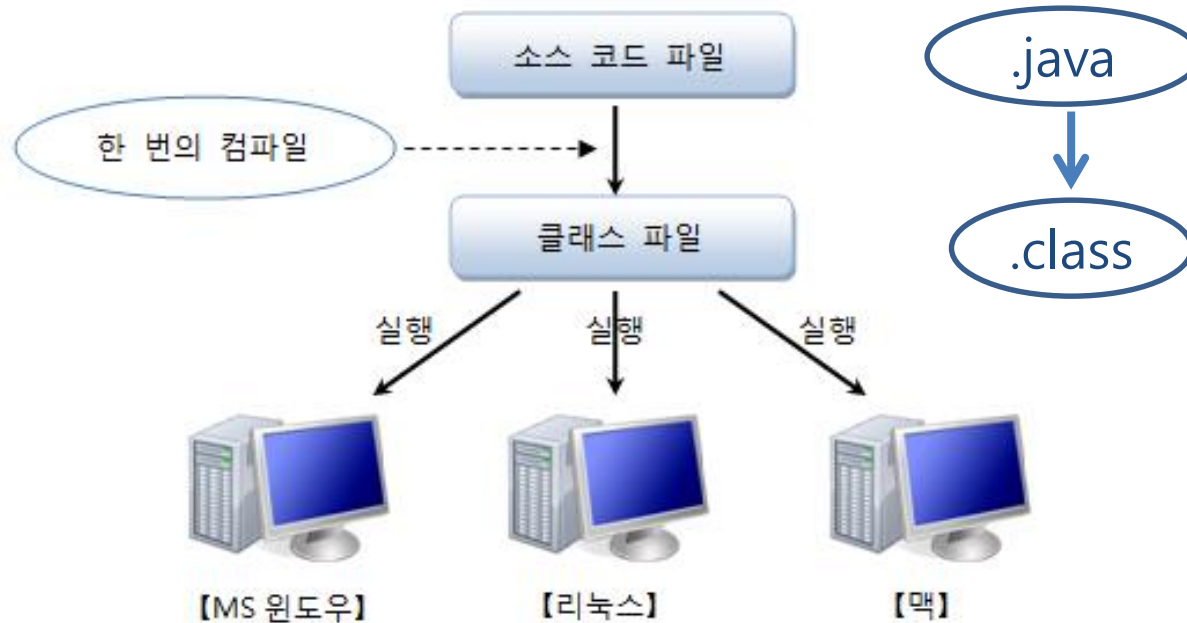


웹 프로그래밍

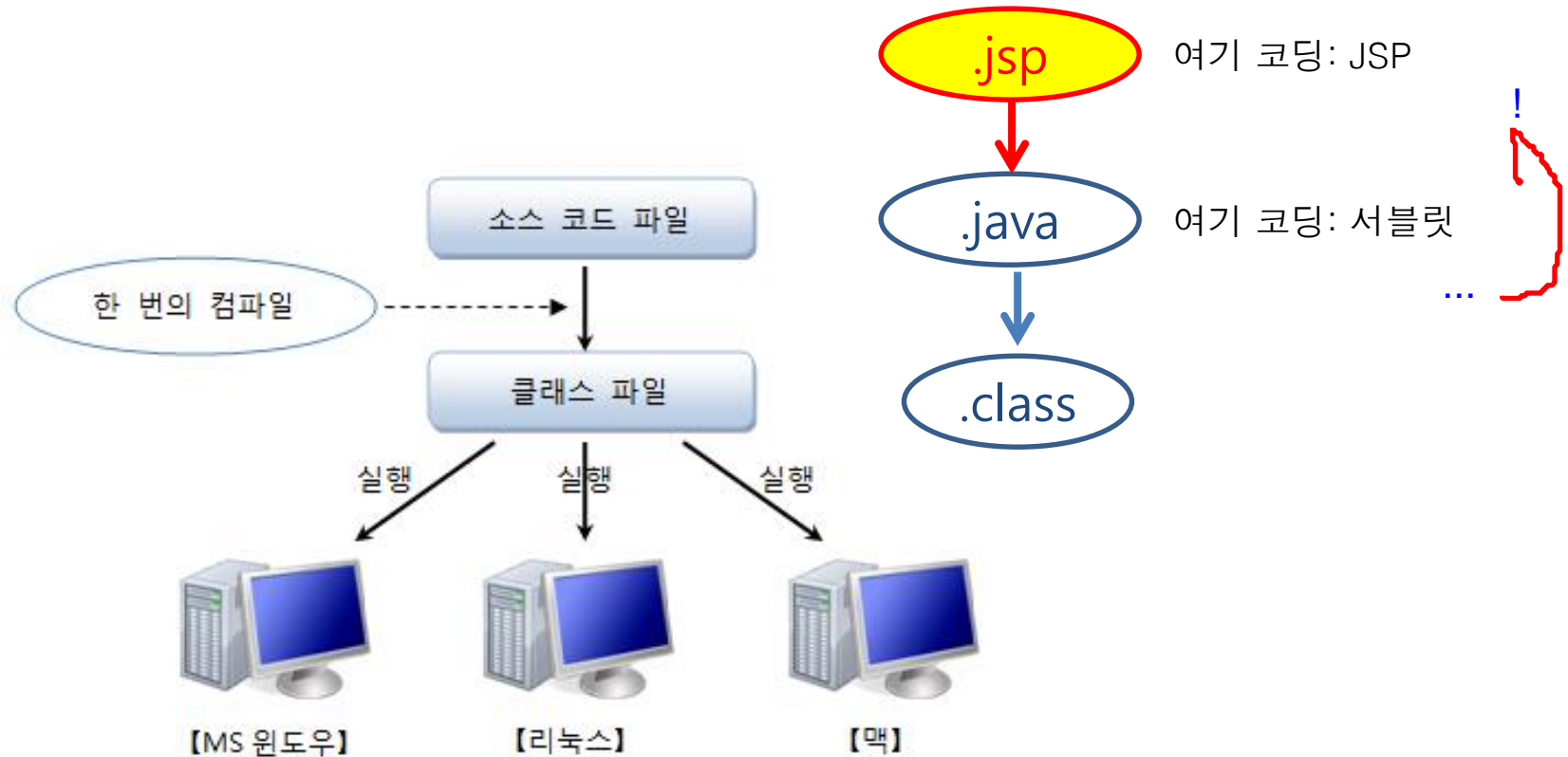
1장. 서블릿과 JSP 개요

동아대학교 컴퓨터공학과

<참고> 1학기 때 배운 자바

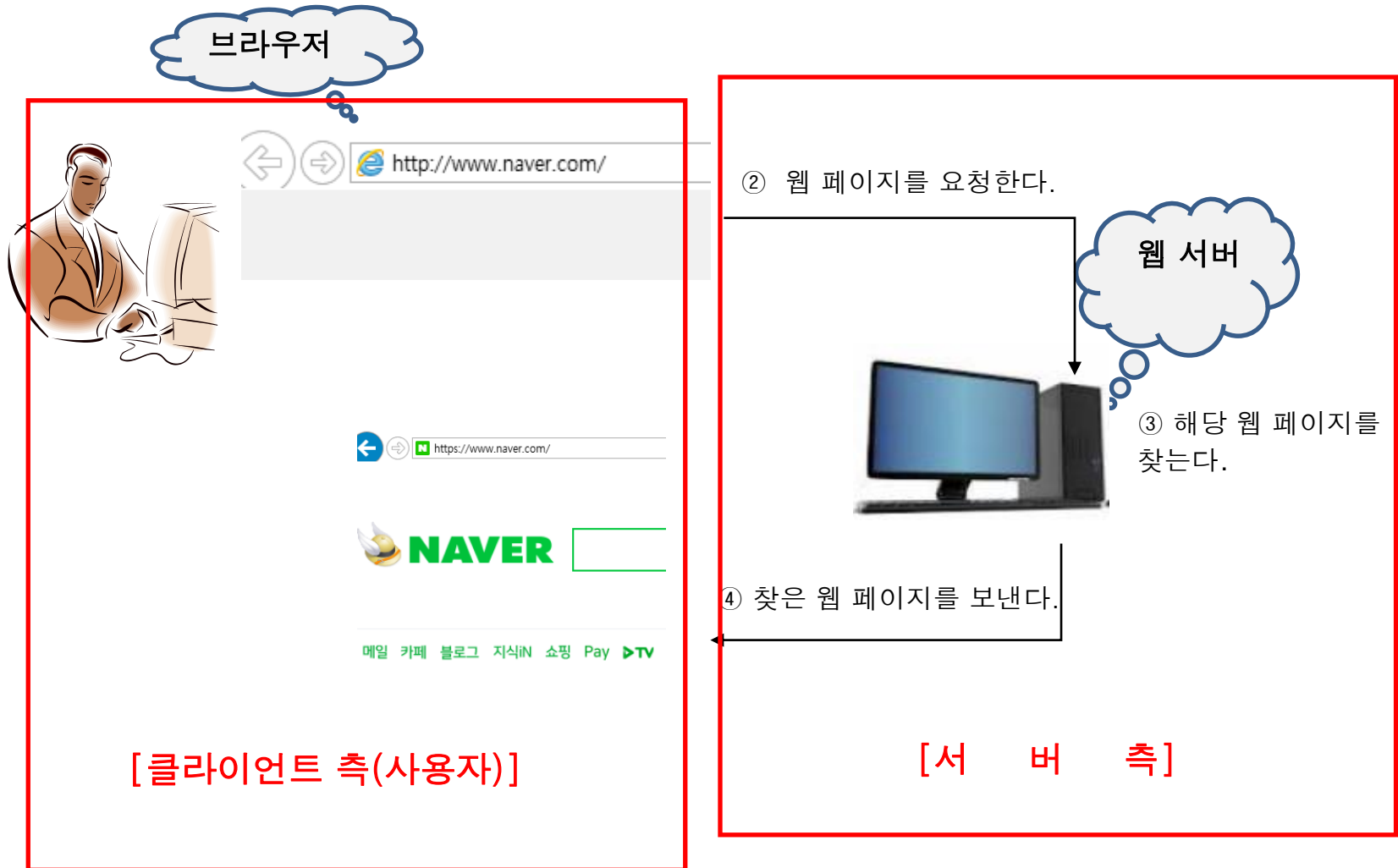


<참고> 2학기 때 배우는 JSP (Java Server Page)



- ❖ 웹 애플리케이션은 웹(인터넷)을 기반으로 실행되는 애플리케이션을 말합니다. 즉, 브라우저로 접근하여 사용되는 응용 소프트웨어를 의미
- ❖ 이 책에서 학습하게 되는 웹 프로그래밍(Web Programming)이 바로 웹 애플리케이션을 제작하는 과정을 뜻합니다.

웹 애플리케이션의 동작원리



HTML

- ❖ 브라우저를 통해서 각종 정보를 제공해주는 웹 페이지는 HTML를 사용하여 웹 프로그래밍을 한 것입니다.

HTML

- ❖ 하지만 HTML만으로는 매일 매일 변경되는 새로운 정보들을 제공해주지 못합니다. 왜냐하면 HTML은 **같은 내용만 표시**해주는 **정적 페이지**이기 때문입니다.
- ❖ 우리가 사용하는 인터넷은 매일 매일 새로운 내용을 제공해주어야 하기 때문에 HTML만 가지고 웹 프로그래밍을 하는데 문제가 있습니다.

서블릿/JSP

- ❖ 그래서 등장하게 된 것이 **동적 페이지**입니다.
 - 서블릿/JSP, PHP, ASP, 요즘엔 파이썬으로도 웹프 많이 함
- ❖ 동적 페이지에서 새로운 정보를 제공해주기 위해서는 방대한 정보를 관리할 데이터베이스가 필요합니다.
 - 예를 들어 게시판에 게재되는 글은 데이터베이스에 저장되었다가 보여주는 것입니다.
- ❖ 이렇듯 다양한 정보를 데이터베이스에서 얻거나 저장해야 하기 위해서 등장한 언어가 PHP, ASP, 서블릿/JSP입니다.

JDK 설치 (1) p. 23

■ 일단 JDK가 이미 설치되었는지 확인

- 시작버튼 → cmd → javac → 뭐라뭐라 길게 나오면 설치된 것임

■ JDK(Java Development Kit) 설치

- 직접 파일 다운로드 받으려면 아래 주소 →

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Java SE Development Kit 8u144		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input checked="" type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.89 MB	jdk-8u144-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u144-linux-arm64-vfp-hflt.tar.gz
Linux x86	164.65 MB	jdk-8u144-linux-i586.rpm
Linux x86	179.44 MB	jdk-8u144-linux-i586.tar.gz
Linux x64	162.1 MB	jdk-8u144-linux-x64.rpm
Linux x64	176.92 MB	jdk-8u144-linux-x64.tar.gz
Mac OS X	226.6 MB	jdk-8u144-macosx-x64.dmg
Solaris SPARC 64-bit	139.87 MB	jdk-8u144-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.18 MB	jdk-8u144-solaris-sparcv9.tar.gz
Solaris x64	140.51 MB	jdk-8u144-solaris-x64.tar.Z
Solaris x64	96.99 MB	jdk-8u144-solaris-x64.tar.gz
Windows x86	190.94 MB	jdk-8u144-windows-i586.exe
Windows x64	197.78 MB	jdk-8u144-windows-x64.exe

JDK 설치 (2)

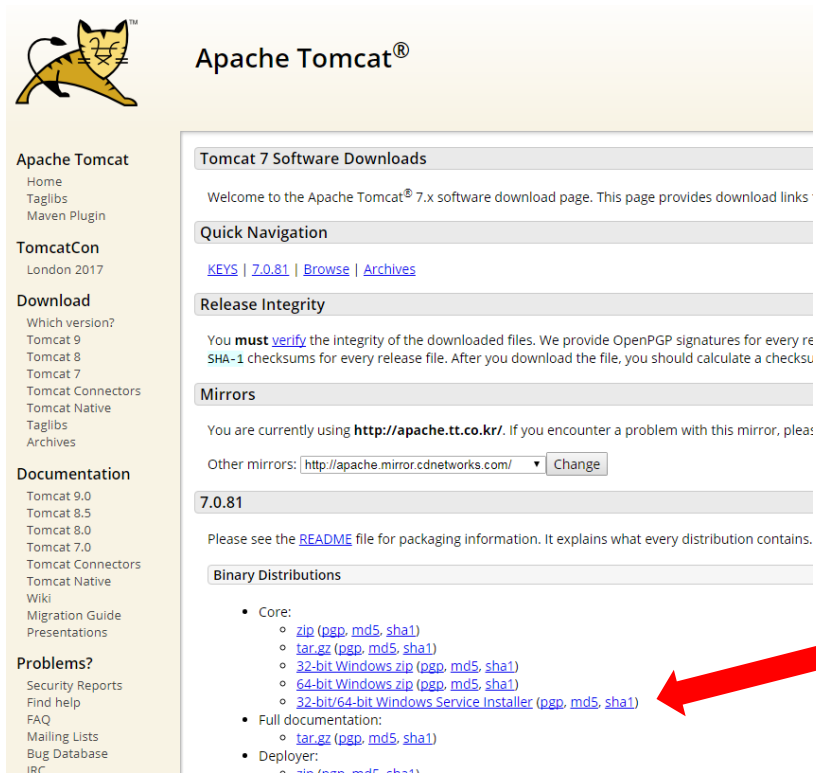
❖ 환경변수

- 설치한 jdk가 jdk-10.0.1 인 경우
- 아래의 3개 시스템변수를 넣으세요. (컴퓨터→시스템속성→고급시스템설정→ 환경변수→ 시스템변수)
- JAVA_HOME → C:\Program Files\Java\jdk-10.0.1
- Path → 맨 앞에 %JAVA_HOME%\bin; 추가
 - 세미콜론 빼먹지 말고
- Classpath → .;%JAVA_HOME%\lib\tools.jar
 - 맨 앞에 마침표랑 세미콜론 넣어주고

톰캣 설치 (1)

■ 톰캣 (Tomcat)

- 직접 파일 다운로드 받으려면 → <http://tomcat.apache.org/download-70.cgi> 에서 7.0버전 다운로드 (더 최신버전 받아도 되는데, 그냥 7.0으로 통일)



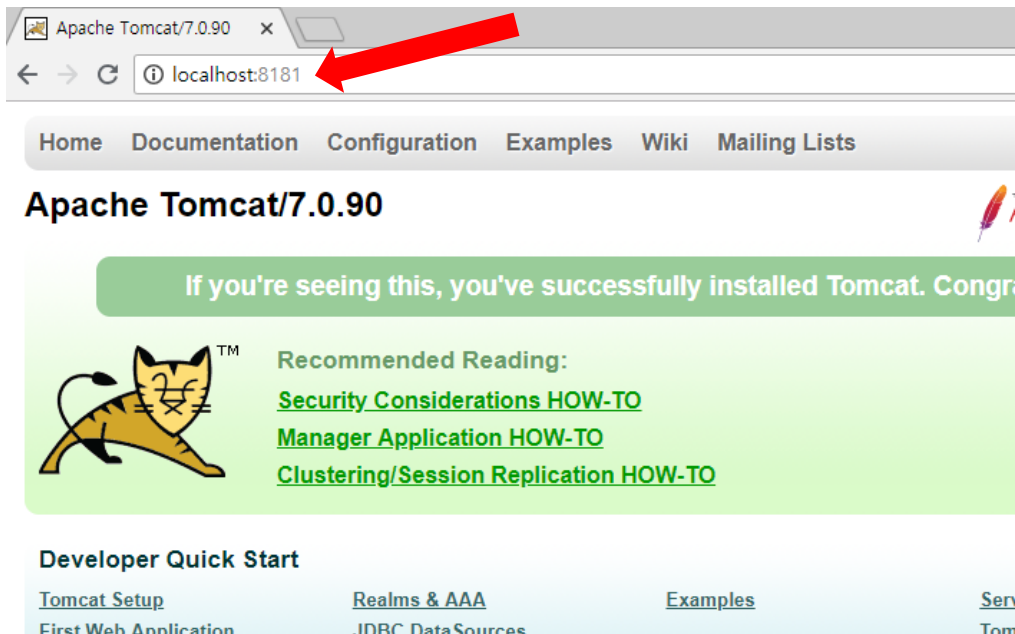
중간에 8080을
8181로 변경

사용자 admin
비번 1234

톰캣 설치 (2)

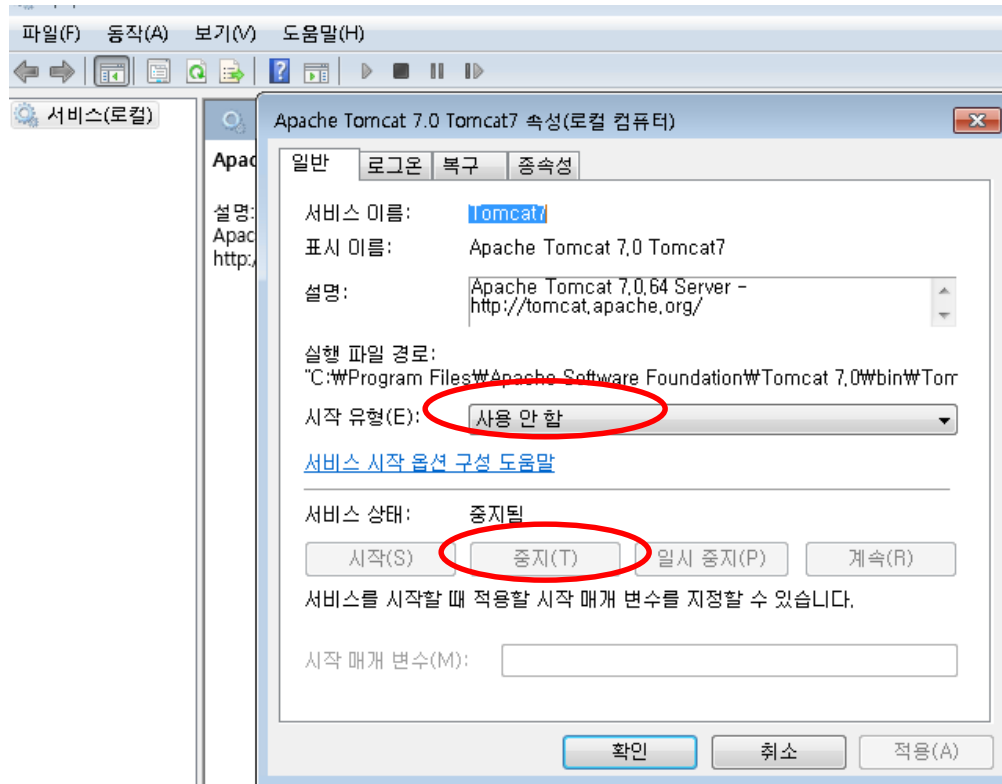
정상작동 여부 확인

브라우저 주소창에 'localhost:8181' 입력하여 톰캣 정상작동 여부 확인



톰캣 설치 (3)

- 예전에는 사람이 직접 톰캣 제어. 지금은 이클립스가 하도록 함
- 이클립스 연동을 위해 [제어판]-관리도구-서비스 → 아파치톰캣 선택 후
- → 중지(T)버튼 누르고 → 시작유형은 사용안함 설정 → 확인



중지 이후에는
localhost:8181 해도
앞 페이지 그림 안 보임
(안 보이는 게 맞으니
걱정마세요)

이클립스 EE 설치 (1)

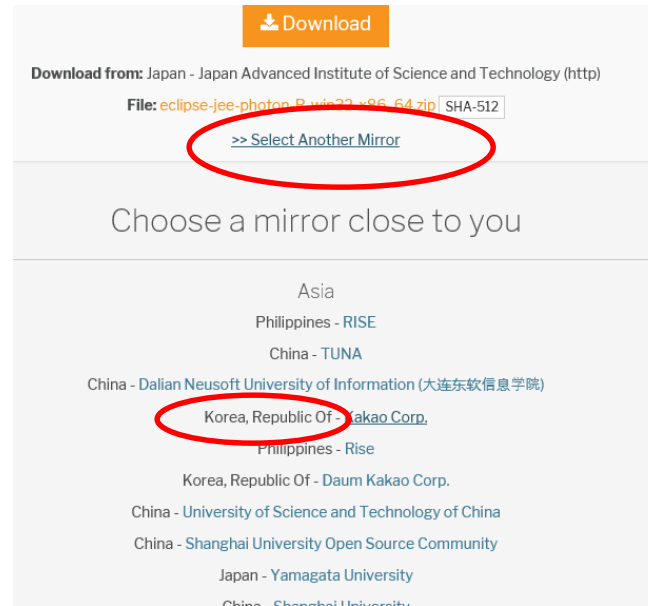
■ 이클립스 EE

- 직접 파일 다운로드 받으려면 → <http://www.eclipse.org/downloads/packages/>



- Eclipse IDE for Java **EE** Developers 버전 설치
- 나중에 이클립스 어디 있나 찾기 편하도록 C:\Program Files\Java 폴더에 zip파일 받은 후 압축해제

이클립스 EE 설치 (2)

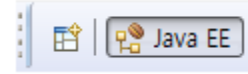


- Select Another Mirror 클릭 후 이왕이면 우리나라 꺼 선택
- 혹시 기다려봐도 자동으로 다운로드 시작 안 하면
땡큐 밑에 작은 click here 선택하면 수동 다운로드 됨 (zip파일)
- 압축만 풀면 끝

이클립스 EE 열었을 때

❖ 퍼스펙티브(Perspective)

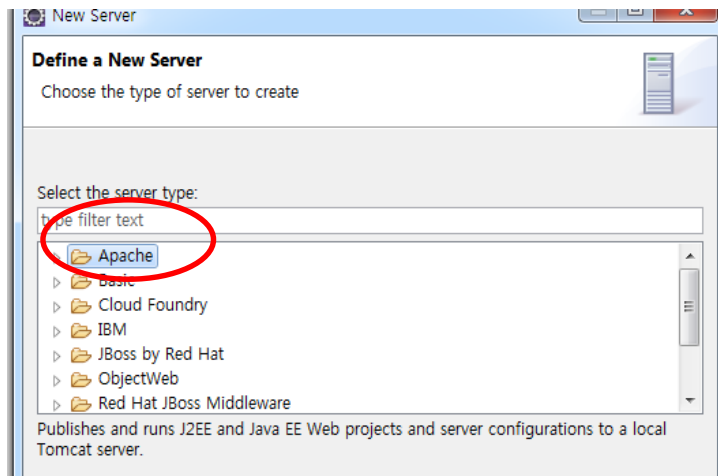
- 화면 우측 상단
- 개발 프로젝트 종류별로 유용한 View들을 묶어놓은 것
- Eclipse IDE for Java EE Developers
 - 기본적으로 Java EE 퍼스펙티브
 - 여기서는 자바 프로그램 코딩은 안 됨
 - 자바 코딩하고 싶으면 클릭 후 Java 퍼스펙티브 선택로 변경



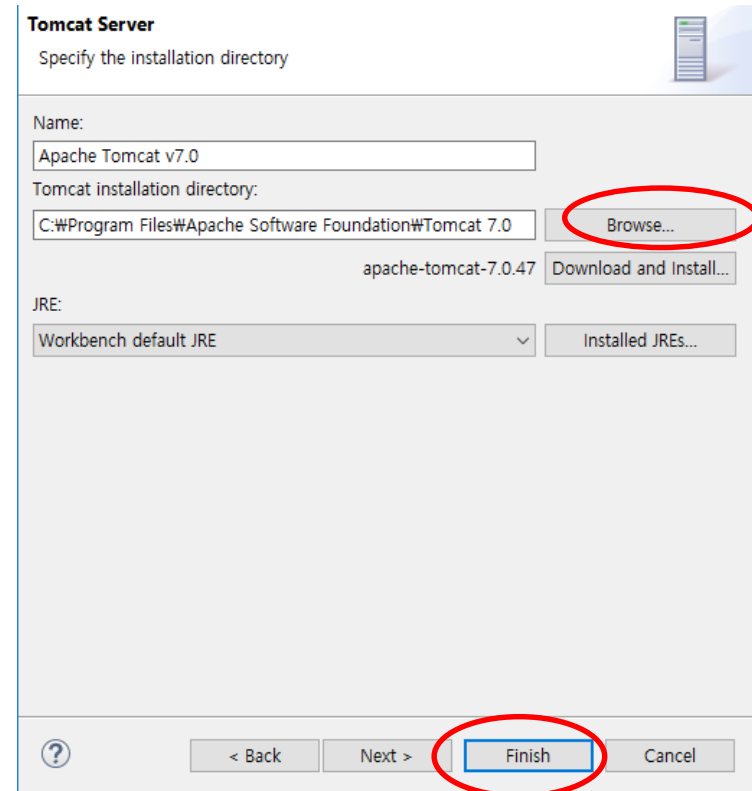
❖ 이클립스에서 Tomcat 서버 지정

(1) 상단메뉴의 [File]-[new]-[other]-[server]밑에 server]-를 통해 톰캣 경로 지정

(2) 아파치 선택 후 톰캣7 선택



(3) 톰캣7 설치된 경로 선택

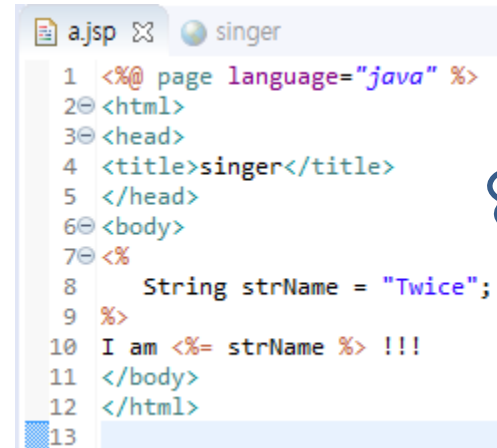


❖ 이클립스에서 프로젝트 생성

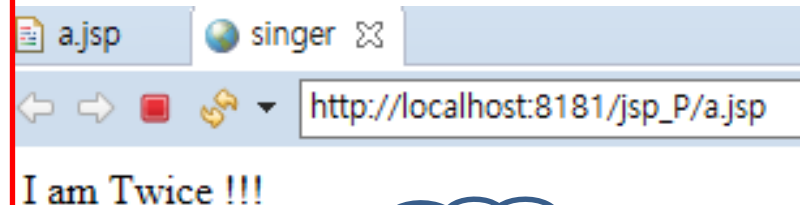
- [File]-[New]-[Dynamic Web Project] 프로젝트 이름 적고 target runtime이 톰캣인지 확인
- 프로젝트이름 오른쪽 마우스 - new - JSP file 선택 후 소스이름 a 입력
- 아래 소스 넣고 실행해보세요~

```
<%@ page language="java" %>
<html>
<head>
<title>singer</title>
</head>
<body>
<%
    String strName = "Twice";
%>
I am <%= strName %> !!!
</body>
</html>
```

jsp!



JSP소스

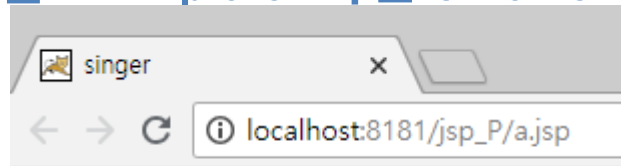


브라우저 결과

❖ 메뉴의 [Window] - [Web Browser]에서 기본 창 선택 가능

- 결과를 이클립스 안(internal)에서 봐도 되고,
- iExplorer 혹은 chrome에서 바로 볼 수도 있음
- (우리는 그냥 default인 internal을 사용하는 걸로)

❖ internal 말고 iExplorer 혹은 chrome으로 선택 후 a.jsp 수행해보세요



브라우저 결과

I am Twice !!!

❖ 페이지 소스보기 해보세요

```
1
2 <html>
3 <head>
4 <title>singer</title>
5 </head>
6 <body>
7
8 I am Twice !!!
9 </body>
10 </html>
11
```

페이지 소스 (HTML)

<참고> internal에서
페이지소스보기 하면 메모장이 뜸

❖ 방화벽 창 뜨면 [엑세스 허용] 누르면 됨

❖ 화면이 잘 나오나요? 그럼 JSP 공부할 환경이 되었습니다!!

- 만약 주소가 localhost가 아니라 실제로 도메인 등록된 거라면
- 전세계에서 내가 만든 웹화면 볼 수 있다는 얘기죠...^^

❖ 실행할 때 팝업창 뜨고 port 사용중 뭐라뭐라 오류 계속 나오면, 명령 프롬프트 창에서

- `netstat -a -n -o -p tcp` 입력해서 포트 확인 ^{port} !
- PID가 4444 경우 `taskkill /f /pid 4444` 입력해서 강제 종료시킴

❖ 한글 들어있어서 폰트 어찌고 하는 문제 나오면

- 그냥 `save as UTF-8` 버튼 선택하면 됨
- 한글 깨지면 jsp 소스 맨 윗줄에 설정
`<%@ page language="java" contentType="text/html; charset=UTF-8"%>`
- html 경우 `<head>~</head>` 사이에 `<meta charset="UTF-8">` 추가로 설정

이클립스에서 인코딩 방식을 UTF-8로 변경하기

- ❖ [Window → Preferences] 메뉴를 선택하여 [Preferences] 창이 나타나면 화면 왼쪽에서는 [General] 하위 항목으로 [Workspace]를 선택하고 화면 오른쪽 상세 내용 중 맨 하단에서는 [Text file encoding] 항목의 선택박스에서 **UTF-8**로 선택합니다.
- ❖ [Preferences] 창이 나타나면 화면 왼쪽에서는 [Web] 하위 항목으로 [JSP Files]를 선택하고 화면 오른쪽 상세 내용 중 [Encoding] 항목의 선택박스에서 **ISO 10646/Unicode(UTF-8)**를 선택합니다.
 - [HTML Files]도 마찬가지로

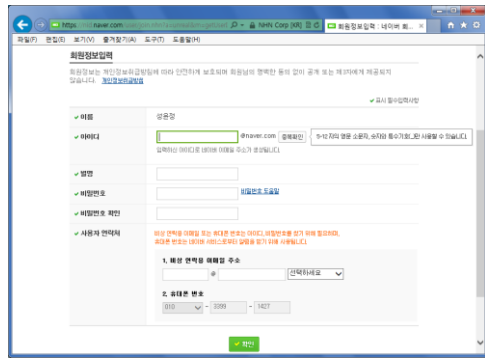
웹 서버(Web Server)/

웹 애플리케이션 서버(Web Application Server : WAS)

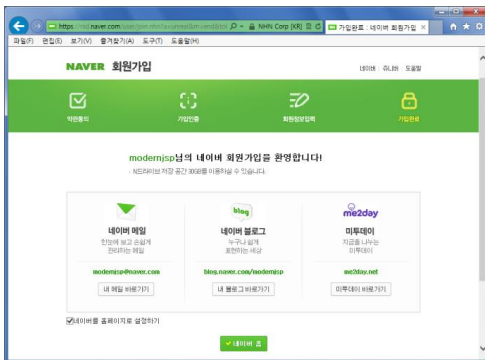
- ❖ 서버는 일반적으로 사용자(클라이언트)의 요청이 들어오면 이를 받아들이어서 결과 페이지를 전송하는 **웹 서버(Web Server)**와 실질적으로 요청한 페이지의 로직이나 데이터베이스와의 연동을 처리할 수 있는 비즈니스 로직이 구현되어야 하는 **웹 애플리케이션 서버(Web Application Server : WAS)**로 이루어져 있습니다.
- ❖ 내 컴퓨터에 웹 애플리케이션 서버인 톰캣을 설치하면 내 컴퓨터는 웹 서비스가 가능한 웹 서버가 됩니다. (톰캣은 웹 서버 역할도 하는 WAS)

?

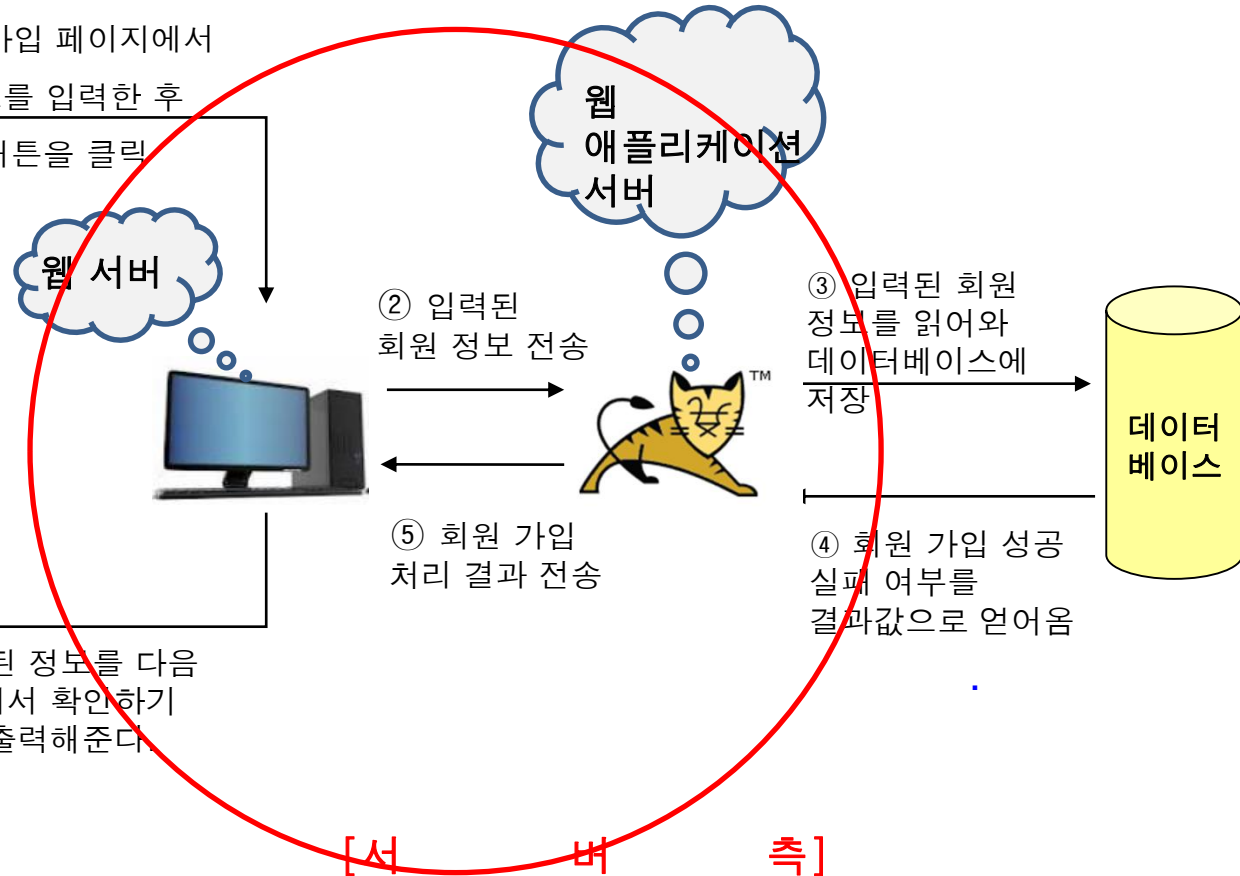
(가)



① 회원 가입 페이지에서
회원 정보를 입력한 후
<확인> 버튼을 클릭



⑥ 입력된 정보를 다음
페이지에서 확인하기
위해서 출력해준다




[클라이언트 측(사용자)]

[서버 측]

요청을 받아서 웹 페이지를 찾아서 보내주는 일을 하는 소프트웨어 (혹은 그 소프트웨어가 설치된 컴퓨터)를 **웹 서버**라고 하고, 요청된 페이지를 응답 받아보는 브라우저를 **클라이언트**라고 합니다.

서블릿(Servlet) p.47

- ❖ 서블릿(Servlet)은 **Server + Applet**의 합성어로 서버 상에서 실행되는 Applet이란 의미로 자바를 이용하여 웹 상에서 실행되는 프로그램을 작성하는 기술을 말합니다.
- ❖ **JSP로 코딩해도 WAS가 서블릿으로 변환한 후 실행됨**

- ❖ 그럼 JSP든 서블릿이든 하나만 알면 되지, 왜 둘 다 배워야 할까요?

서블릿 예제

```
import javax.servlet.http.HttpServlet;
import java.io.PrintWriter;
public class AdditonServlet01 extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        int num1 = 20;
        int num2 = 10;
        int add = num1 + num2;
        PrintWriter out = response.getWriter();
        out.println("<html> <head> <title>Additon</title> </head>");
        out.println("<body>");
        out.println(num1 + "+" + num2 + "=" + add);
        out.println("</body>");
        out.println("</html>");
    }
}
```



!



```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<head> <title>Addition</title> </head>
<body>
<%
    int num1 = 20;
    int num2 = 10;
    int add = num1 + num2;
%>
<%=num1%> + <%=num2%> = <%=add%>
</body>
</html>
```